

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

- 1 1. (Currently amended) A method for ~~providing object change information from a first system to a second system for synchronizing the second system with the first system, the second system having an object cache for storing objects~~ synchronizing object caches, the method comprising: ~~the steps of:~~
2 changing ~~a~~ an ~~first~~ object in the first system, a second object in a second system, and a third object in a third system, wherein each object contains at least one unchanged attribute;
3
4
5
6
7
8 determining a first object change set, a second object change set, and a third object change set, -which represents the changes made to the first object, the second object, and the third object, respectively, and wherein an object change set contains an attribute if and only if the attribute has been changed; in the first system; and
9
10
11
12
13 sending the first object change set from the first system directly to the second system ~~to which causes~~ the second system to use a second merging unit to apply the first object change set to the corresponding object in the second system; ~~s cache so as to synchronize the second system with the first system,~~ wherein control is returned to a client coupled to the first system whether or not the second system has completed processing the object change set, wherein the second system is registered in the first system prior to sending the object change information from the first system to the second system, and wherein the
14
15
16
17
18
19
20

21 registration process causes the first system to know that the second system wants
22 to receive object change information of objects changed in the first system,
23 thereby eliminating the need for the second system to register with every object it
24 is interested in

25 sending the second object change set from the second system directly to
26 the third system which causes the third system to use a third merging unit to apply
27 the second object change set to the corresponding object in the third system; and
28 sending the third object change set from the third system directly to the
29 first system which causes the first system to use a first merging unit to apply the
30 third object change set to the corresponding object in the first system.

1 2. (Currently amended) The method as claimed in claim 1 further
2 comprising a step of establishing a communication link between the first system
3 and the second system wherein the ~~distributing~~ ~~sending~~ step ~~distributes~~ ~~sends~~ the
4 first object change information-set from the first system to the second system
5 through the communication link.

1 3. (Original) The method as claimed in claim 2 wherein the establishing
2 step establishes the communication link based on a publish/subscribe protocol.

1 4. (Canceled).

1 5. (Currently amended) The method as claimed in claim 1 further
2 comprising a step of sending the first object change information-set to a database
3 for updating the corresponding object in the database ~~with the object change~~
4 ~~information~~.

1 6. (Currently amended) The method as claimed in claim 5 further
2 comprising the steps of:
3 receiving an error message from the database ~~when-if~~ the updating of the
4 corresponding object in the database fails; and
5 discarding the first object change information-set prior to the ~~distributing~~
6 sending step in response to the error message.

1 7. (Canceled).

1 8. (Canceled).

1 9. (Canceled).

1 10. (Canceled).

1 11. (Canceled).

1 12. (Currently amended) A method for ~~providing object change~~
2 ~~information from a first system to a second system for synchronizing the second~~
3 ~~system with the first system, the first system having a first object cache for storing~~
4 ~~one or more objects and the second system having a second object cache for~~
5 ~~storing one or more objects~~ synchronizing object caches, the method comprising:
6 the steps of:
7 receiving a first object change set at a first system, a second object change
8 set at a second system, and a third object change set at a third system, which
9 represent the changes made to a first object on the first system, a second object on
10 the second system, and a third object on the third system, respectively;

11 wherein each object contains at least one unchanged attribute, and wherein
12 an object change set contains an attribute if and only if the attribute has been
13 changed;

14 sending the first object change set from the first system directly to the
15 second system which causes the second system to use a second merging unit to
16 apply the first object change set to the corresponding object in the second system;
17 sending the second object change set from the second system directly to
18 the third system which causes the third system to use a third merging unit to apply
19 the second object change set to the corresponding object in the third system; and
20 sending the third object change set from the third system directly to the
21 first system which causes the first system to use a first merging unit to apply the
22 third object change set to the corresponding object in the first system.~~determining~~
23 ~~object change sets which represent changes made to objects in the first system;~~
24 and

25 ~~sending the object change sets from the first system to the second system~~
26 ~~to cause the second system to apply the object change sets to corresponding~~
27 ~~objects in the second object cache so as to synchronize the objects in the second~~
28 ~~cache of the second system with the changed objects in the first system, wherein~~
29 ~~control is returned to a client coupled to the first system whether or not the second~~
30 ~~system has completed processing the object change set, wherein the second~~
31 ~~system is registered in the first system prior to sending the object change~~
32 ~~information from the first system to the second system, and where the registration~~
33 ~~process causes the first system to know that the second system wants to receive~~
34 ~~object change information of objects changed in the first system, thereby~~
35 ~~eliminating the need for the second system to register with every object it is~~
36 ~~interested in.~~

1 13. (Currently amended) The method as claimed in claim 12 further
2 comprising a step of establishing a communication link between the first system
3 and the second system wherein the ~~distributing~~sending step ~~distributes~~sends the
4 first object change ~~information~~set from the first system to the second system
5 through the communication link.

1 14. (Canceled).

1 15. (Currently amended) The method as claimed in claim 12 further
2 comprising a step of sending the first object change ~~information~~set from the first
3 system to a database for updating the corresponding object in the database ~~with~~
4 ~~the object change information~~.

1 16. (Currently amended) The method as claimed in claim 15 further
2 comprising the steps of:
3 receiving an error message from the database ~~when~~if the updating
4 of the corresponding object in the database fails; and
5 discarding the first object change ~~information~~set prior to the
6 ~~distributing~~sending step in response to the error message.

1 17. (Canceled).

1 18. (Canceled).

1 19. (Canceled).

1 20. (Canceled).

1 21. (Canceled).

1 22. (Canceled).

1 23. (Canceled).

1 24. (Canceled).

1 25. (Canceled).

1 26. (Canceled).

1 27. (Canceled).

1 28. (Canceled).

1 29. (Canceled).

1 30. (Canceled).

1 31. (Canceled).

1 32. (Currently amended) A Computer computer-readable storage-media
2 medium storing instructions for use in the execution in a computer ofthat when
3 executed by a computer cause the computer to perform a method for providing
4 object change information from a first system to a second system for
5 synchronizing the second system with the first system, the second system having

6 an object cache for storing objectssynchronizing object caches, the method
7 comprising: the steps of:
8 receiving a first object change set at a first system, a second object change
9 set at a second system, and a third object change set at a third system, which
10 represent the changes made to a first object on the first system, a second object on
11 the second system, and a third object on the third system, respectively;
12 wherein each object contains at least one unchanged attribute, and wherein
13 an object change set contains an attribute if and only if the attribute has been
14 changed;
15 sending the first object change set from the first system directly to the
16 second system which causes the second system to use a second merging unit to
17 apply the first object change set to the corresponding object in the second system;
18 sending the second object change set from the second system directly to
19 the third system which causes the third system to use a third merging unit to apply
20 the second object change set to the corresponding object in the third system; and
21 sending the third object change set from the third system directly to the
22 first system which causes the first system to use a first merging unit to apply the
23 third object change set to the corresponding object in the first system.~~changing an~~
24 ~~object in the first system;~~
25 determining an object change set which represents changes made to the
26 object in the first system;
27
28 sending the object change set from the first system to the second system to
29 cause the second system to apply the object change set to the corresponding object
30 in the second system's cache so as to synchronize the second system with the first
31 system, wherein control is returned to a client coupled to the first system whether
32 or not the second system has completed processing the object change set, wherein

33 | the second system is registered in the first system prior to sending the object
34 | change information from the first system to the second system, and wherein the
35 | registration process causes the first system to know that the second system wants
36 | to receive object change information of objects changed in the first system;
37 | thereby eliminating the need for the second system to register with every object it
38 | is interested in.

1 33. (Canceled).

1 34. (Canceled).